Renesas Electronics America Inc - UPD78F1144AGB-GAH-AX Datasheet

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	78K/0R
Core Size	16-Bit
Speed	20MHz
Connectivity	3-Wire SIO, I²C, LINbus, UART/USART
Peripherals	DMA, LVD, POR, PWM, WDT
Number of I/O	50
Program Memory Size	128KB (128K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-LQFP
Supplier Device Package	-
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/upd78f1144agb-gah-ax

3.3.4 Register direct addressing

[Function]

Register direct addressing stores in the program counter (PC) the contents of a general-purpose register pair (AX/BC/DE/HL) and CS register of the current register bank specified with the instruction word as 20-bit data, and specifies the program address. Register direct addressing can be applied only to the CALL AX, BC, DE, HL, and BR AX instructions.

Figure 3-22. Outline of Register Direct Addressing

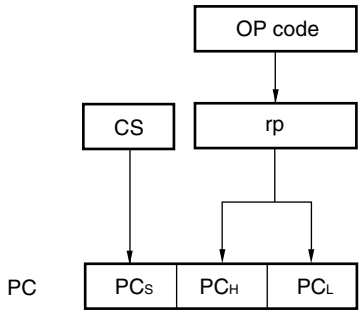
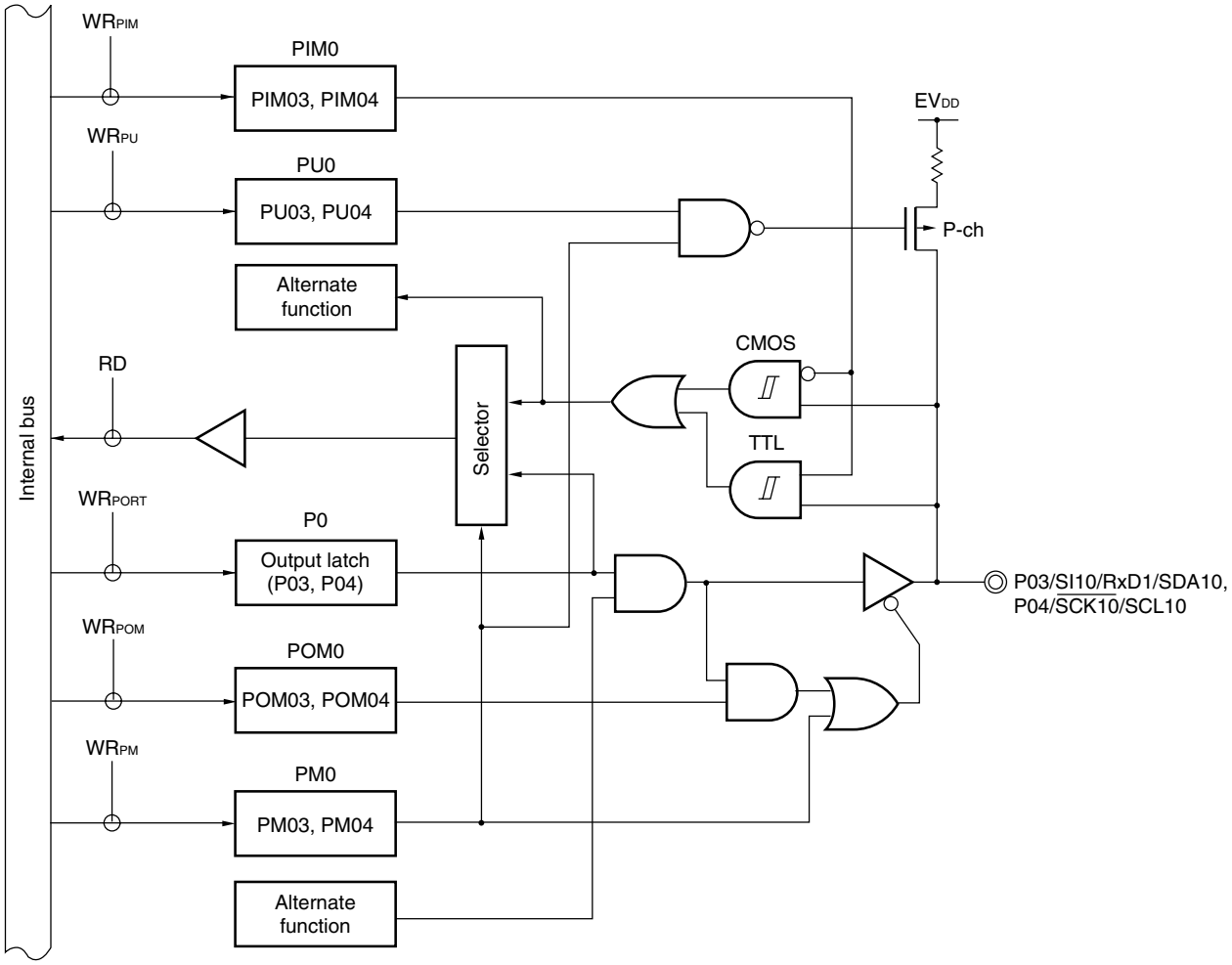


Figure 4-5. Block Diagram of P03 and P04



P0: Port register 0
PU0: Pull-up resistor option register 0
PM0: Port mode register 0
PIM0: Port input mode register 0
POM0: Port output mode register 0
RD: Read signal
WR××: Write signal

4.2.2 Port 1

Port 1 is an 8-bit I/O port with an output latch. Port 1 can be set to the input mode or output mode in 1-bit units using port mode register 1 (PM1). When the P10 to P17 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 1 (PU1).

This port can also be used for external interrupt request input, serial interface data I/O, clock I/O, timer I/O, and real-time counter clock output.

Reset signal generation sets port 1 to input mode.

Figures 4-7 to 4-11 show block diagrams of port 1.

Cautions 1. To use P10/$\overline{\text{SCK00}}$ P11/SI00/RxD0 P12/SO00/TxD0, P13/TxD3, or P14/RxD3 as a general-purpose port, note the serial array unit setting. For details, refer to the following tables.

- **Table 11-5 Relationship Between Register Settings and Pins (Channel 0 of Unit 0: CSI00, UART0 Transmission)**
- **Table 11-6 Relationship Between Register Settings and Pins (Channel 1 of Unit 0: UART0 Reception)**
- **Table 11-9 Relationship Between Register Settings and Pins (Channel 2 of Unit 1: UART3 Transmission)**
- **Table 11-10 Relationship Between Register Settings and Pins (Channel 3 of Unit 1: UART3 Reception)**

2. To use P16/TI01/TO01/INTP5 or P17/TI02/TO02 as a general-purpose port, set bits 1 and 2 (TO01, TO02) of timer output register 0 (TO0) and bits 1 and 2 (TOE01, TOE02) of timer output enable register 0 (TOE0) to "0", which is the same as their default status setting.

3. To use P15/RTCDIV/RTCCL as a general-purpose port, set bit 4 (RCLOE0) of real-time counter control register 0 (RTCC0) and bit 6 (RCLOE2) of real-time counter control register 2 (RTCC2) to "0", which is the same as their default status settings.

4.2.6 Port 5

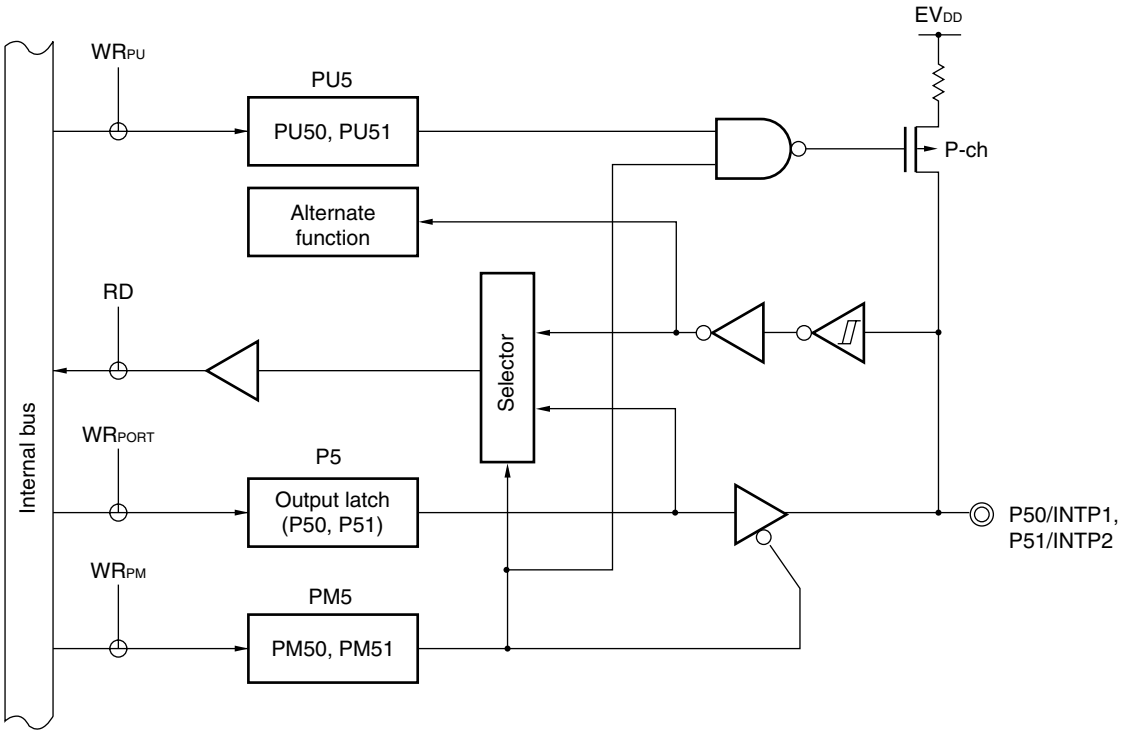
Port 5 is an 6-bit I/O port with an output latch. Port 5 can be set to the input mode or output mode in 1-bit units using port mode register 5 (PM5). When the P50 to P55 pins are used as an input port, use of an on-chip pull-up resistor can be specified in 1-bit units by pull-up resistor option register 5 (PU5).

This port can also be used for external interrupt request input.

Reset signal generation sets port 5 to input mode.

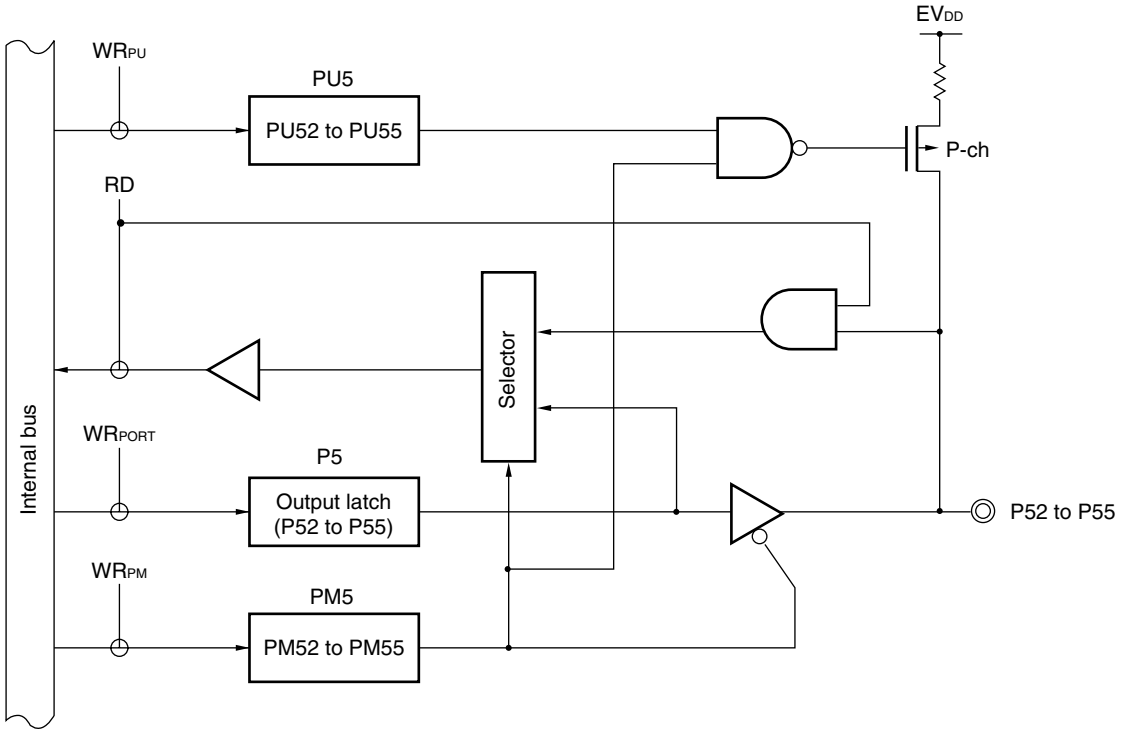
Figures 4-18 and 4-19 show block diagrams of port 5.

Figure 4-18. Block Diagram of P50 and P51



P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
RD: Read signal
WR××: Write signal

Figure 4-19. Block Diagram of P52 to P55



P5: Port register 5
PU5: Pull-up resistor option register 5
PM5: Port mode register 5
RD: Read signal
WR××: Write signal

5.4.3 Internal high-speed oscillator

The internal high-speed oscillator is incorporated in the 78K0R/KE3 (8 MHz (TYP.)). Oscillation can be controlled by bit 0 (HIOSTOP) of the clock operation status control register (CSC).

After a reset release, the internal high-speed oscillator automatically starts oscillation.

5.4.4 Internal low-speed oscillator

The internal low-speed oscillator is incorporated in the 78K0R/KE3.

The internal low-speed oscillation clock is used only as the watchdog timer clock. The internal low-speed oscillation clock cannot be used as the CPU clock.

After a reset release, the internal low-speed oscillator automatically starts oscillation, and the watchdog timer is driven (240 kHz (TYP.)) if the watchdog timer operation is enabled by the option byte.

The internal low-speed oscillator continues oscillation except when the watchdog timer stops. When the watchdog timer operates, the internal low-speed oscillation clock does not stop, even in case of a program loop.

5.4.5 Prescaler

The prescaler generates CPU/peripheral hardware clock by dividing the main system clock and subsystem clock.

Table 5-4. CPU Clock Transition and SFR Register Setting Examples (4/4)

<R> **(9) CPU clock changing from subsystem clock (D) to high-speed system clock (C)**

(Setting sequence of SFR registers) →

Setting Flag of SFR Register / Status Transition	OSTS Register	CSC Register	OSMC Register	OSTC Register	CKC Register	
		MSTOP	FSEL		MCM0	CSS
(D) → (C) (X1 clock: 2 MHz ≤ fx ≤ 10 MHz)	**Note 1**	0	0	Must be checked	1	0
(D) → (C) (X1 clock: 10 MHz < fx ≤ 20 MHz)	**Note 1**	0	1 Note 2	Must be checked	1	0
(D) → (C) (external main clock)	**Note 1**	0	0/1	Must not be checked	1	0

Unnecessary if the CPU is operating with the high-speed system clock

Unnecessary if these registers are already set

Notes 1. Set the oscillation stabilization time as follows.
- Desired OSTC oscillation stabilization time ≤ Oscillation stabilization time set by OSTS

2. FSEL = 1 when f_{CLK} > 10 MHz
If a divided clock is selected and f_{CLK} ≤ 10 MHz, use with FSEL = 0 is possible even if fx > 10 MHz.

Caution Set the clock after the supply voltage has reached the operable voltage of the clock to be set (see CHAPTER 27 ELECTRICAL SPECIFICATIONS (STANDARD PRODUCTS) and CHAPTER 28 ELECTRICAL SPECIFICATIONS ((A) GRADE PRODUCTS)).

(10) • HALT mode (E) set while CPU is operating with internal high-speed oscillation clock (B)
• HALT mode (F) set while CPU is operating with high-speed system clock (C)
• HALT mode (G) set while CPU is operating with subsystem clock (D)

Status Transition	Setting
(B) → (E) (C) → (F) (D) → (G)	Executing HALT instruction

<R> **(11) • STOP mode (H) set while CPU is operating with internal high-speed oscillation clock (B)**
• STOP mode (I) set while CPU is operating with high-speed system clock (C)

(Setting sequence) →

Status Transition		Setting		
(B) → (H)		Stopping peripheral functions that cannot operate in STOP mode	–	Executing STOP instruction
(C) → (I)	In X1 oscillation		Sets the OSTS register	
	External clock		–	

Remark (A) to (I) in Table 5-4 correspond to (A) to (I) in Figure 5-15.

Figure 6-22. Format of Noise Filter Enable Register 1 (NFEN1)

Address: F0061H After reset: 00H R/W

Symbol	7	6	5	4	3	2	1	0
NFEN1	0	TNFEN06	TNFEN05	TNFEN04	TNFEN03	TNFEN02	TNFEN01	TNFEN00

TNFEN06	Enable/disable using noise filter of TI06/TO06/P06 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN05	Enable/disable using noise filter of TI05/TO05/P05 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN04	Enable/disable using noise filter of TI04/TO04/P42 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN03	Enable/disable using noise filter of TI03/TO03/INTP4/P31 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN02	Enable/disable using noise filter of TI02/TO02/P17 pin input signal
0	Noise filter OFF
1	Noise filter ON

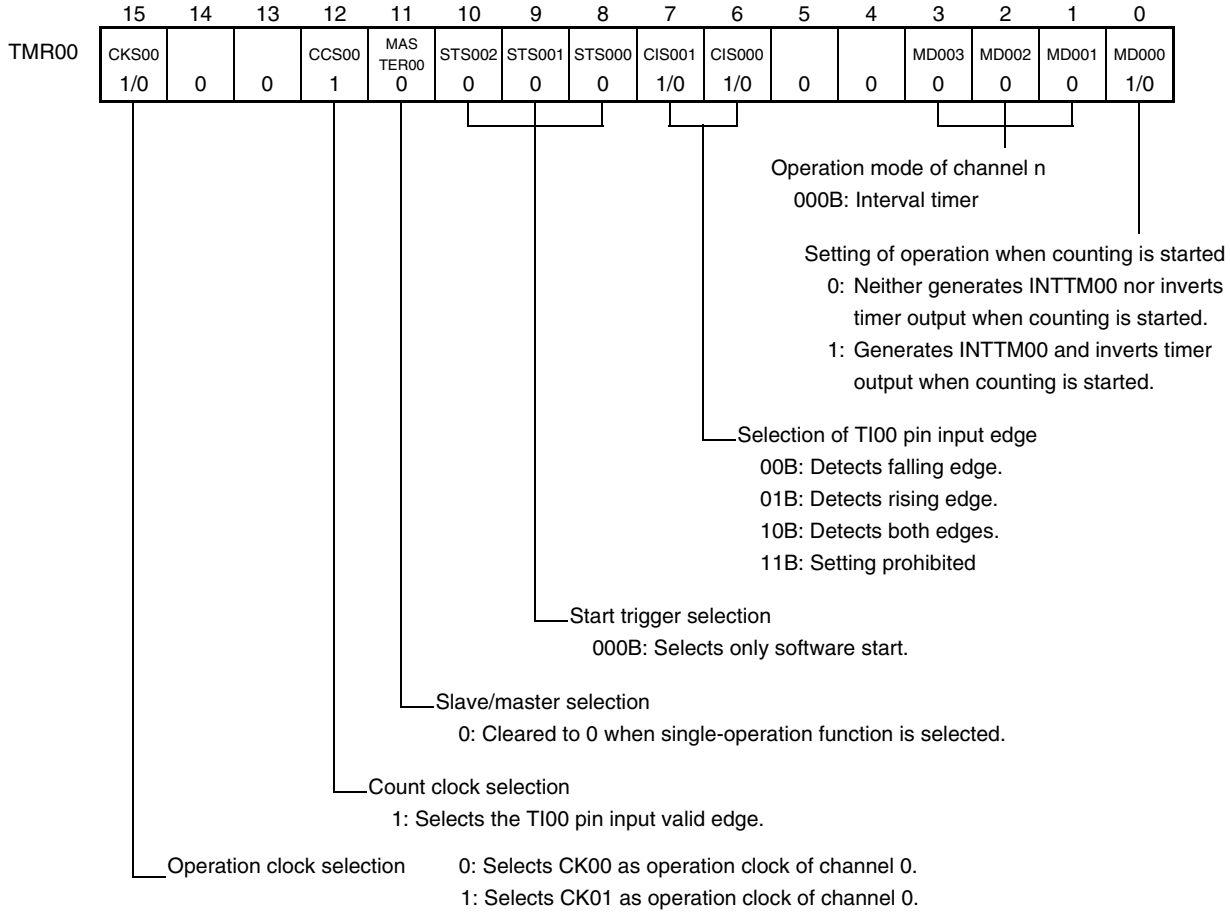
TNFEN01	Enable/disable using noise filter of TI01/TO01/INTP5/P16 pin input signal
0	Noise filter OFF
1	Noise filter ON

TNFEN00	Enable/disable using noise filter of TI00/P00 pin input signal
0	Noise filter OFF
1	Noise filter ON

Caution Be sure to clear bits 7 to "0".

Figure 6-45. Example of Set Contents of Registers When Frequency Divider Is Used

(a) Timer mode register 00 (TMR00)



(b) Timer output register 0 (TO0)

	Bit 0	
TO0	TO00 1/0	0: Outputs 0 from TO00. 1: Outputs 1 from TO00.

(c) Timer output enable register 0 (TOE0)

	Bit 0	
TOE0	TOE00 1/0	0: Stops the TO00 output operation by counting operation. 1: Enables the TO00 output operation by counting operation.

(d) Timer output level register 0 (TOL0)

	Bit 0	
TOL0	TOL00 0	0: Cleared to 0 when TOM00 = 0 (toggle mode)

(e) Timer output mode register 0 (TOM0)

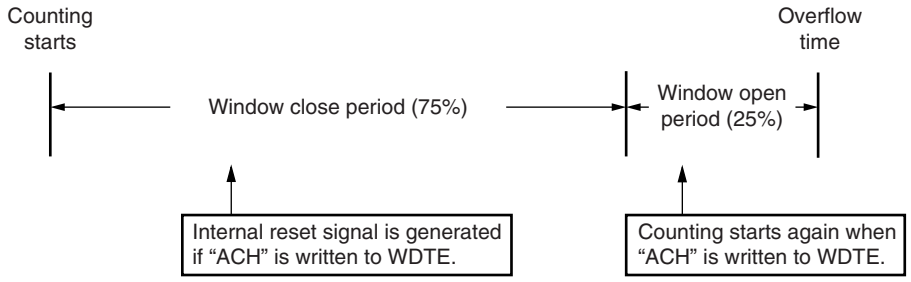
	Bit 0	
TOM0	TOM00 0	0: Sets toggle mode.

8.4.3 Setting window open period of watchdog timer

Set the window open period of the watchdog timer by using bits 6 and 5 (WINDOW1, WINDOW0) of the option byte (000C0H). The outline of the window is as follows.

- If “ACH” is written to WDTE during the window open period, the watchdog timer is cleared and starts counting again.
- Even if “ACH” is written to WDTE during the window close period, an abnormality is detected and an internal reset signal is generated.

Example: If the window open period is 25%



Caution When data is written to WDTE for the first time after reset release, the watchdog timer is cleared in any timing regardless of the window open time, as long as the register is written before the overflow time, and the watchdog timer starts counting again.

The window open period to be set is as follows.

Table 8-4. Setting Window Open Period of Watchdog Timer

WINDOW1	WINDOW0	Window Open Period of Watchdog Timer
0	0	25%
0	1	50%
1	0	75%
1	1	100%

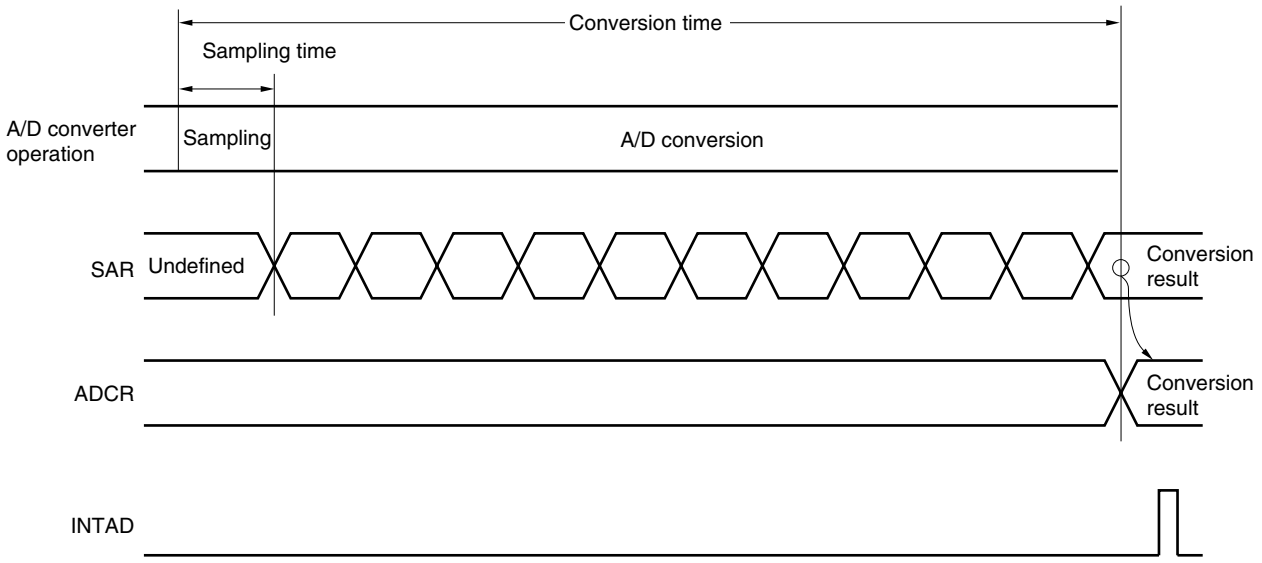
Cautions 1. The watchdog timer continues its operation during self-programming of the flash memory and EEPROM emulation. During processing, the interrupt acknowledge time is delayed. Set the overflow time and window size taking this delay into consideration.

2. When bit 0 (WDSTBYON) of the option byte (000C0H) = 0, the window open period is 100% regardless of the values of WINDOW1 and WINDOW0.

3. Do not set the window open period to 25% if the watchdog timer corresponds to either of the conditions below.

- **When used at a supply voltage (V_{DD}) below 2.7 V.**
- **When stopping all main system clocks (internal high-speed oscillation clock, X1 clock, and external main system clock) by use of the STOP mode or software.**
- **Low-power consumption mode**

Figure 10-12. Basic Operation of A/D Converter



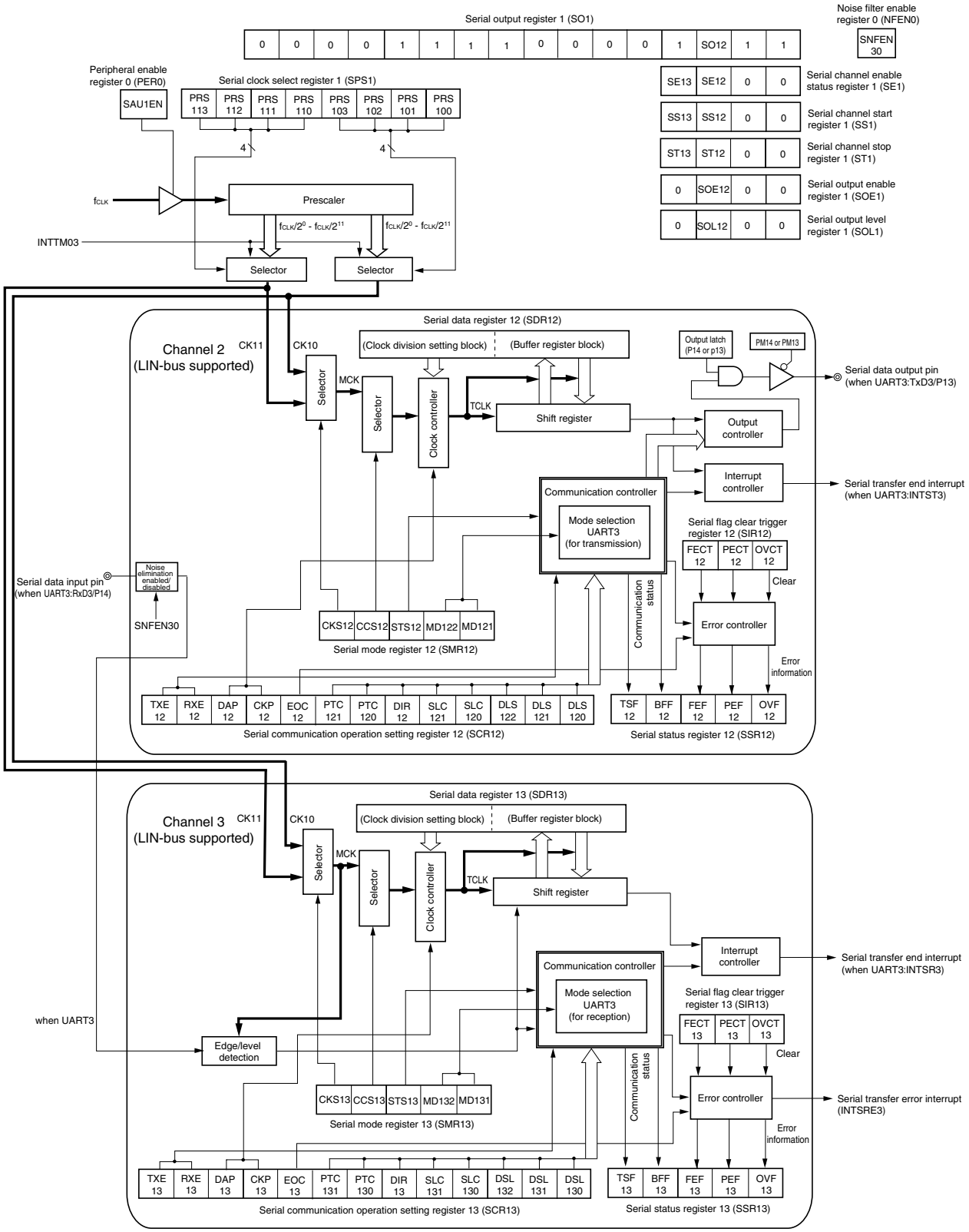
A/D conversion operations are performed continuously until bit 7 (ADCS) of the A/D converter mode register (ADM) is reset (0) by software.

If a write operation is performed to the analog input channel specification register (ADS) during an A/D conversion operation, the conversion operation is initialized, and if the ADCS bit is set (1), conversion starts again from the beginning.

Reset signal generation clears the A/D conversion result register (ADCR, ADCRH) to 0000H or 00H.

Figure 11-2 shows the block diagram of serial array unit 1.

Figure 11-2. Block Diagram of Serial Array Unit 1



(4) Serial communication operation setting register mn (SCRmn)

SCRmn is a communication operation setting register of channel n. It is used to set a data transmission/reception mode, phase of data and clock, whether an error signal is to be masked or not, parity bit, start bit, stop bit, and data length.

Rewriting SCRmn is prohibited when the register is in operation (when SEmn = 1).

SCRmn can be set by a 16-bit memory manipulation instruction.

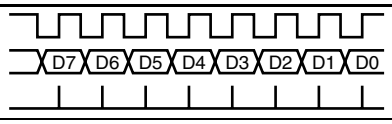
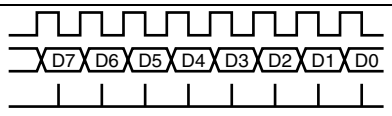
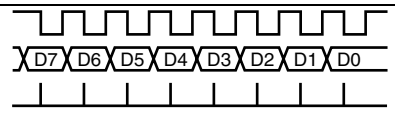
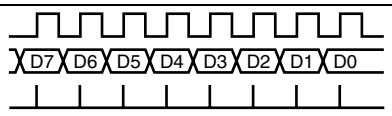
Reset signal generation sets this register to 0087H.

Figure 11-7. Format of Serial Communication Operation Setting Register mn (SCRmn) (1/3)

Address: F0118H, F0119H (SCR00) to F011EH, F011FH (SCR03), F015CH, F015DH (SCR12), F015EH, F015FH (SCR13) After reset: 0087H R/W

Symbol	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCRmn	TXE mn	RXE mn	DAP mn	CKP mn	0	EOC mn	PTC mn1	PTC mn0	DIR mn	0	SLC mn1	SLC mn0	0	DLS mn2	DLS mn1	DLS mn0

TXE mn	RXE mn	Setting of operation mode of channel n
0	0	Does not start communication.
0	1	Reception only
1	0	Transmission only
1	1	Transmission/reception

DAP mn	CKP mn	Selection of data and clock phase in CSI mode	Type
0	0		1
0	1		2
1	0		3
1	1		4
Be sure to set DAPmn, CKPmn = 0, 0 in the UART mode and simplified I^2C mode.			

Caution Be sure to clear bits 3, 6, and 11 to "0". Be sure to set bit 2 to "1".

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0 to 3), mn = 00 to 03, 12, 13, p: CSI number (p = 00, 10)

(2) Baud rate error during transmission

The baud rate error of UART (UART0, UART1, UART3) communication during transmission can be calculated by the following expression. Make sure that the baud rate at the transmission side is within the permissible baud rate range at the reception side.

(Baud rate error) = (Calculated baud rate value) ÷ (Target baud rate) × 100 − 100 [%]

Here is an example of setting a UART baud rate at f_{CLK} = 20 MHz.

UART Baud Rate (Target Baud Rate)	f_{CLK} = 20 MHz			
	Operation Clock (MCK)	SDRmn[15:9]	Calculated Baud Rate	Error from Target Baud Rate
300 bps	$f_{CLK}/2^9$	64	300.48 bps	+0.16 %
600 bps	$f_{CLK}/2^8$	64	600.96 bps	+0.16 %
1200 bps	$f_{CLK}/2^7$	64	1201.92 bps	+0.16 %
2400 bps	$f_{CLK}/2^6$	64	2403.85 bps	+0.16 %
4800 bps	$f_{CLK}/2^5$	64	4807.69 bps	+0.16 %
9600 bps	$f_{CLK}/2^4$	64	9615.38 bps	+0.16 %
19200 bps	$f_{CLK}/2^3$	64	19230.8 bps	+0.16 %
31250 bps	$f_{CLK}/2^3$	39	31250.0 bps	±0.0 %
38400 bps	$f_{CLK}/2^2$	64	38461.5 bps	+0.16 %
76800 bps	$f_{CLK}/2$	64	76923.1 bps	+0.16 %
153600 bps	f_{CLK}	64	153846 bps	+0.16 %
312500 bps	f_{CLK}	31	312500 bps	±0.0 %

Remark m: Unit number (m = 0, 1), n: Channel number (n = 0, 2), mn = 00, 02, 12

Figure 12-6. Format of IIC Control Register 0 (IICC0) (1/4)

Address: FFF52H After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	<3>	<2>	<1>	<0>
IICC0	IICE0	LREL0	WREL0	SPIE0	WTIM0	ACKE0	STT0	SPT0

IICE0	I^2C operation enable
0	Stop operation. Reset IIC status register 0 (IICS0)[Note 1]. Stop internal operation.
1	Enable operation.
Be sure to set this bit (1) while the SCL0 and SDA0 lines are at high level.	
Condition for clearing (IICE0 = 0)	Condition for setting (IICE0 = 1)
• Cleared by instruction • Reset	• Set by instruction

LREL0[Note 2]	Exit from communications
0	Normal operation
1	This exits from the current communications and sets standby mode. This setting is automatically cleared to 0 after being executed. Its uses include cases in which a locally irrelevant extension code has been received. The SCL0 and SDA0 lines are set to high impedance. The following flags of IIC control register 0 (IICC0) and IIC status register 0 (IICS0) are cleared to 0. • STT0 • SPT0 • MSTS0 • EXC0 • COI0 • TRC0 • ACKD0 • STD0
The standby mode following exit from communications remains in effect until the following communications entry conditions are met. • After a stop condition is detected, restart is in master mode. • An address match or extension code reception occurs after the start condition.	
Condition for clearing (LREL0 = 0)	Condition for setting (LREL0 = 1)
• Automatically cleared after execution • Reset	• Set by instruction

WREL0[Note 2]	Wait cancellation
0	Do not cancel wait
1	Cancel wait. This setting is automatically cleared after wait is canceled.
When WREL0 is set (wait canceled) during the wait period at the ninth clock pulse in the transmission status (TRC0 = 1), the SDA0 line goes into the high impedance state (TRC0 = 0).	
Condition for clearing (WREL0 = 0)	Condition for setting (WREL0 = 1)
• Automatically cleared after execution • Reset	• Set by instruction

Notes 1. The IICS0 register, the STCF and IICBSY bits of the IICF0 register, and the CLD0 and DAD0 bits of the IICCL0 register are reset.

2. The signal of this bit is invalid while IICE0 is 0.

Caution The start condition is detected immediately after I^2C is enabled to operate (IICE0 = 1) while the SCL0 line is at high level and the SDA0 line is at low level. Immediately after enabling I^2C to operate (IICE0 = 1), set LREL0 (1) by using a 1-bit memory manipulation instruction.

(1) DMA mode control register n (DMCn)

DMCn is a register that is used to set a transfer mode of DMA channel n. It is used to select a transfer direction, data size, setting of pending, and start source. Bit 7 (STGn) is a software trigger that starts DMA.

Rewriting bits 6, 5, and 3 to 0 of DMCn is prohibited during operation (when DSTn = 1).

DMCn can be set by a 1-bit or 8-bit memory manipulation instruction.

Reset signal generation clears this register to 00H.

Figure 14-4. Format of DMA Mode Control Register n (DMCn) (1/2)

Address: FFFBAH (DMC0), FFFBBH (DMC1) After reset: 00H R/W

Symbol	<7>	<6>	<5>	<4>	3	2	1	0
DMCn	STGn	DRSn	DSn	DWAITn	IFCn3	IFCn2	IFCn1	IFCn0

STGn[Note 1]	DMA transfer start software trigger
0	No trigger operation
1	DMA transfer is started when DMA operation is enabled (DENn = 1).
DMA transfer is performed once by writing 1 to STGn when DMA operation is enabled (DENn = 1). When this bit is read, 0 is always read.	

DRSn	Selection of DMA transfer direction
0	SFR to internal RAM
1	Internal RAM to SFR

DSn	Specification of transfer data size for DMA transfer
0	8 bits
1	16 bits

DWAITn[Note 2]	Pending of DMA transfer
0	Executes DMA transfer upon DMA start request (not held pending).
1	Holds DMA start request pending if any.
DMA transfer that has been held pending can be started by clearing the value of DWAITn to 0. It takes 2 clocks to actually hold DMA transfer pending when the value of DWAITn is set to 1.	

Notes 1. The software trigger (STGn) can be used regardless of the IFCn0 to IFCn3 values.

<R> **2.** When DMA transfer is held pending while using both DMA channels, be sure to hold the DMA transfer pending for both channels (by setting DWAIT0 and DWAIT1 to 1).

Remark n: DMA channel number (n = 0, 1)

14.5 Example of Setting of DMA Controller

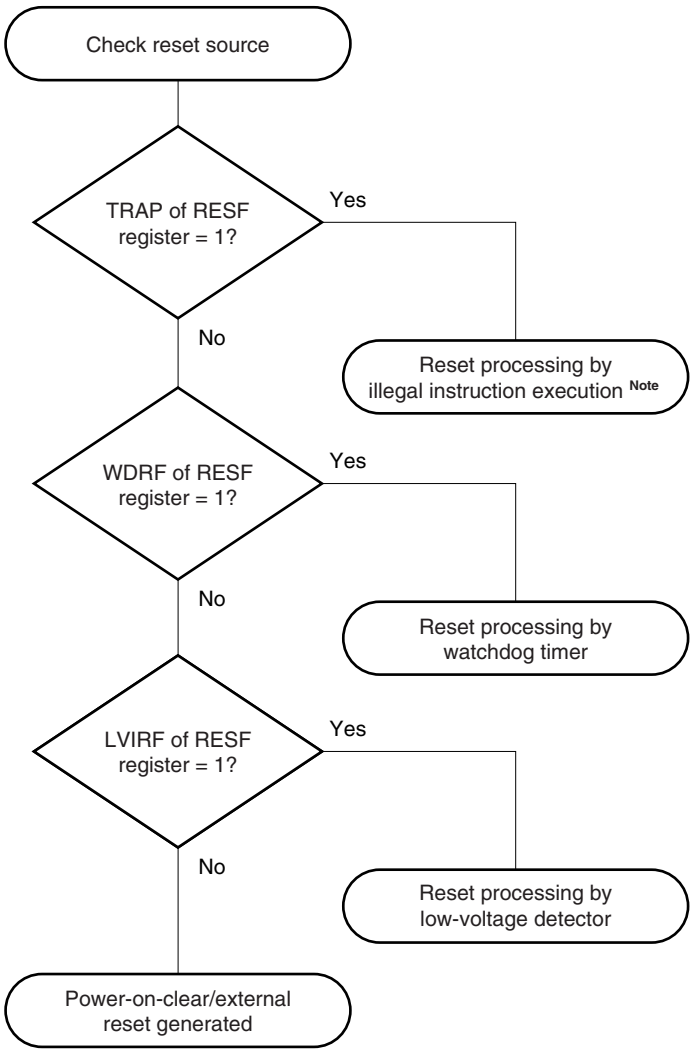
14.5.1 CSI consecutive transmission

A flowchart showing an example of setting for CSI consecutive transmission is shown below.

- Consecutive transmission (256 bytes) of CSI00
- DMA channel 0 is used for DMA transfer.
- DMA start source: INTCSI00 (software trigger (STG0) only for the first start source)
- Interrupt of CSI00 is specified by IFC03 to IFC00 (bits 3 to 0 of the DMC0 register) = 0110B.
- <R> Transfers FF100H to FF1FFH (256 bytes) of RAM to FFF10H of the data register (SIO00) of CSI.

Figure 19-3. Example of Software Processing After Reset Release (2/2)

• Checking reset source



Note The illegal instruction is generated when instruction code FFH is executed.
Reset by the illegal instruction execution not issued by emulation with the in-circuit emulator or on-chip debug emulator.

23.9 Flash Memory Programming by Self-Programming

The 78K0R/KE3 supports a self-programming function that can be used to rewrite the flash memory via a user program. Because this function allows a user application to rewrite the flash memory by using the 78K0R/KE3 self-programming library, it can be used to upgrade the program in the field.

If an interrupt occurs during self-programming, self-programming can be temporarily stopped and interrupt servicing can be executed. If an unmasked interrupt request is generated in the EI state, the request branches directly from the self-programming library to the interrupt routine. After the self-programming mode is later restored, self-programming can be resumed. However, the interrupt response time is different from that of the normal operation mode.

Remark For details of the self-programming function and the 78K0R/KE3 self-programming library, refer to **78K0R Microcontroller Self Programming Library Type01 User's Manual (U18706E)**.

Cautions
1. **The self-programming function cannot be used when the CPU operates with the subsystem clock.**
2. **In the self-programming mode, call the self-programming start library (FlashStart).**
3. **To prohibit an interrupt during self-programming, in the same way as in the normal operation mode, execute the self-programming library in the state where the IE flag is cleared (0) by the DI instruction. To enable an interrupt, clear (0) the interrupt mask flag to accept in the state where the IE flag is set (1) by the EI instruction, and then execute the self-programming library.**
4. **The self-programming function is disabled in the low consumption current mode. For details of the low consumption current mode, see CHAPTER 21 REGULATOR.**
5. **Disable DMA operation (DENn = 0) during the execution of self programming library functions.**

Chapter	Classification	Function	Details of Function	Cautions	Page
Chapter 3	Soft	Memory space	Internal data memory space	It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space for fetching instructions or as a stack area.	p.56 ☐
				While using the self-programming function, the area of FFE20H to FFEFFH cannot be used as a stack memory. Furthermore, the areas of FCF00H to FD6FFH cannot be used with the μPD78F1146 and 78F1146A.	p.56 ☐
			SFR: Special function register area	Do not access addresses to which SFRs are not assigned.	pp.57, 68 ☐
			2nd SFR: Extended special function register	Do not access addresses to which extended SFR is not assigned.	pp.57, 74 ☐
		Processor registers	SP: Stack pointer	Since reset signal generation makes the SP contents undefined, be sure to initialize the SP before using the stack.	p.64 ☐
				The values of the stack pointer must be set to even numbers. If odd numbers are specified, the least significant bit is automatically cleared to 0.	p.64 ☐
				It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space as a stack area.	p.64 ☐
				While using the self-programming function, the area of FFE20H to FFEFFH cannot be used as a stack memory. Furthermore, the areas of FCF00H to FD6FFH cannot be used with the μPD78F1146 and 78F1146A.	p.64 ☐
			General-purpose registers	It is prohibited to use the general-purpose register (FFEE0H to FFEFFH) space for fetching instructions or as a stack area.	p.65 ☐
Chapter 4	Soft	Port functions	P01/TO00, P05/TI05/TO05, P06/TI06/TO06	To use P01/TO00, P05/TI05/TO05, or P06/TI06/TO06 as a general-purpose port, set bits 0, 5,and 6 (TO00, TO05, TO06) of timer output register 0 (TO0) and bits 0, 5, and 6 (TOE00, TOE05,TOE06) of timer output enable register 0 (TOE0) to “0”, which is the same as their default status setting.	p.96 ☐
			P02/SO10/TxD1, P03/SI10/RxD1/SDA10, P04/$\overline{\text{SCK10}}$/SCL10	To use P02/SO10/TxD1, P03/SI10/RxD1/SDA10, or P04/$\overline{\text{SCK10}}$/SCL10 as a general-purpose port, note the serial array unit 0 setting. For details, refer to the following tables. • Table 11-7 Relationship Between Register Settings and Pins (Channel 2 of Unit 0: CSI10, UART1 Transmission, IIC10) • Table 11-8 Relationship Between Register Settings and Pins (Channel 3 of Unit 0: UART1 Reception)	p.96 ☐
			P10/$\overline{\text{SCK00}}$/, P11/SI00/RxD0, P12/SO00/TxD0 P13/TxD3, P14/RxD3	To use P10/$\overline{\text{SCK00}}$, P11/SI00/RxD0, P12/SO00/TxD0, P13/TxD3, or P14/RxD3 as a general-purpose port, note the serial array unit setting. For details, refer to the following tables. • Table 11-5 Relationship Between Register Settings and Pins (Channel 0 of Unit 0: CSI00, UART0 Transmission) • Table 11-6 Relationship Between Register Settings and Pins (Channel 1 of Unit 0:, UART0 Reception) • Table 11-9 Relationship Between Register Settings and Pins (Channel 2 of Unit 1: UART3 Transmission) • Table 11-10 Relationship Between Register Settings and Pins (Channel 3 of Unit 1: UART3 Reception)	p.101 ☐